REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
APRIL 2009

2. REPORT TYPE
Conference Paper Post Print

3. DATES COVERED *(From - To)*
March 2008-March 2009

4. TITLE AND SUBTITLE
Adversarial Intent Modeling Using Embedded Simulation and Temporal Bayesian Knowledge Bases

5a. CONTRACT NUMBER
FA8750-08-C-0101

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)
Nicholas J. Pioch, James Melhuish, Andy Seidel, Eugene Santos, Jr., Deqing Li, and Mark Gorniak

5d. PROJECT NUMBER
459S

5e. TASK NUMBER
8C

5f. WORK UNIT NUMBER
P1

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BAE Systems Advanced Information Technologies
6 New England Executive Park
Burlington, MA 01803-5012

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RISC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2009-9

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited. PA# 88ABW-2009-1133 Date Cleared: 16-March-2009

13. SUPPLEMENTARY NOTES

14. ABSTRACT
To foster shared battlespace awareness among air strategy planners, BAE Systems has developed Commander's Model Integration and Simulation Toolkit (CMIST), an Integrated Development Environment for authoring, integration, validation, and debugging of models relating multiple domains, including political, military, social, economic and information. CMIST provides a unified graphical user interface for such systems of systems modeling, spanning several disparate modeling paradigms. Here, we briefly review the CMIST architecture and then compare modeling results using two approaches to intent modeling. The first uses reactive agents with simplified behavior models that apply rule-based triggers to initiate actions based solely on observations of the external world at the current time in the simulation. The second method models proactive agents running an embedded CMIST simulation representing their projection of how events may unfold in the future in order to take early preventative action. Finally, we discuss a recent extension to CMIST that incorporates Temporal Bayesian Knowledge Bases for more sophisticated models of adversarial intent that are capable of inferring goals and future actions given evidence of current actions at particular times.

15. SUBJECT TERMS
Adversary modeling, intent inference, Bayesian Knowledge Bases, model integration, agent-based modeling, system dynamics, dynamic Bayesian networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mark J. Gorniak
U	U	U	UU	18	19b. TELEPHONE NUMBER (*Include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Adversarial intent modeling using embedded simulation and temporal Bayesian knowledge bases

Nicholas J. Pioch[a], James Melhuish[a], Andy Seidel[a], Eugene Santos, Jr.[b], Deqing Li[b], Mark Gorniak[c]
[a]BAE Systems Advanced Information Technologies,
[a]6 New England Executive Pk, Burlington, MA, USA 01803;
[a]{nicholas.pioch, james.melhuish, andy.seidel}@baesystems.com
[b]Dartmouth College, Thayer School of Engineering, 8000 Cummings Hall, Hanover, NH, USA 03755
[b]{eugene.santos.jr, deqing.li}@dartmouth.edu}
[c]Air Force Research Laboratory, AFRL/RISB, 525 Brooks Road, Rome NY 13441;
[c]{gorniakm@rl.af.mil}

ABSTRACT

To foster shared battlespace awareness among air strategy planners, BAE Systems has developed Commander's Model Integration and Simulation Toolkit (CMIST), an Integrated Development Environment for authoring, integration, validation, and debugging of models relating multiple domains, including political, military, social, economic and information. CMIST provides a unified graphical user interface for such systems of systems modeling, spanning several disparate modeling paradigms. Here, we briefly review the CMIST architecture and then compare modeling results using two approaches to intent modeling. The first uses *reactive* agents with simplified behavior models that apply rule-based triggers to initiate actions based solely on observations of the external world at the current time in the simulation. The second method models *proactive* agents running an embedded CMIST simulation representing their projection of how events may unfold in the future in order to take early preventative action. Finally, we discuss a recent extension to CMIST that incorporates Temporal Bayesian Knowledge Bases for more sophisticated models of adversarial intent that are capable of inferring goals and future actions given evidence of current actions at particular times.

Keywords: Adversary modeling, intent inference, Bayesian Knowledge Bases, model integration, agent-based modeling, system dynamics, dynamic Bayesian networks

Modeling and Simulation for Military Operations IV, edited by Dawn A. Trevisani, Proc. of SPIE Vol. 7348, 73480E © 2009 SPIE CCC code: 0277-786X/09/$18 doi: 10.1117/12.818926

1. INTRODUCTION

Recent military operations have demonstrated an urgent need for the U.S. Air Force and the joint community to expand the scope of its intelligence, planning, and operations beyond traditional force-on-force strategy against enemy military and infrastructure targets to include political, economic, social, and information considerations as well. Previous attempts in the modeling and simulation community to model such non-kinetic effects have been limited to non-executable conceptual models that are designed to describe but not actually simulate dynamic, complex interactions [1]. Other approaches that seek to model non-kinetic factors as scalar force limiters or multipliers within existing large-scale executable legacy simulation environments offer only limited insight into non-kinetic and kinetic interactions due to lack of explicit non-kinetic process models and actual feedback to the kinetic models [2]. To address this problem, under the AFRL Commander's Predictive Environment program, BAE Systems Advanced Information Technologies developed the Commander's Model Integration and Simulation Toolkit (CMIST) as an integrated exploratory modeling environment combining multiple simulation paradigms such as system dynamics, Bayesian cause-effect models, and agent-based discrete event models, to enable rapid forecasting of battlespace effects (see Figure 1). For technical details on CMIST, see [3] and [4].

Through the first two years of development, we have applied CMIST's hybrid modeling environment to develop strategic and operational models of varying scale for notional scenarios, including a small-scale counter-insurgency vignette and a larger-scale air- and ground-based combat operation in which CMIST was used to compare effects across alternative courses of action with vs. without initial PSYOPS missions [5]. Both of these modeling efforts used a simplified approach to agent intent modeling in which an agent compares one or more observed properties of the outside world to specified thresholds and initiates particular actions by changing one or more variables in the world model under its control. We refer to this as *reactive intent modeling* in the sense that the agents are simply responding to world events as they occur. Recently, we have extended CMIST to allow an agent to run an embedded CMIST simulation model representing its own internal, possibly simplified, model of the outside world. This *proactive intent model* allows the agent to project the future state of the world, including adversary actions, in order to take appropriate preventative measures. We discuss in Sections 0 and 0 comparative results using reactive vs. proactive intent models in the context of our notional PSYOPS-Combat scenario.

We encountered two limitations of the rule-based agent modeling family used in these modeling experiments. First, the intent models lacked explicit representation of the goals and beliefs driving an agent's behavior. Second, the models lacked a mechanism for reasoning over evidence to infer which goals an agent is currently pursuing as well as future actions implied by those goals. To address these limitations, BAE Systems collaborated with Dartmouth College to incorporate Bayesian Knowledge Bases (BKB) as a new CMIST modeling family for adversary intent modeling. Details on the technical approach to BKB are found in [6] and [7], and recent applications to modeling and wargaming are discussed in [8] and [9]. In Section 0 we describe an example of a BKB intent model used to infer whether an adversary's course of action involves conventional warfare, irregular warfare, or a combination. We then discuss in Section 0 ongoing work by Dartmouth to extend this capability to support temporal constraints between belief nodes in a BKB that can be used to infer times of future actions based on the time and certainty of observed evidence. We preface these discussions with a brief overview of CMIST's system architecture in Section 0.

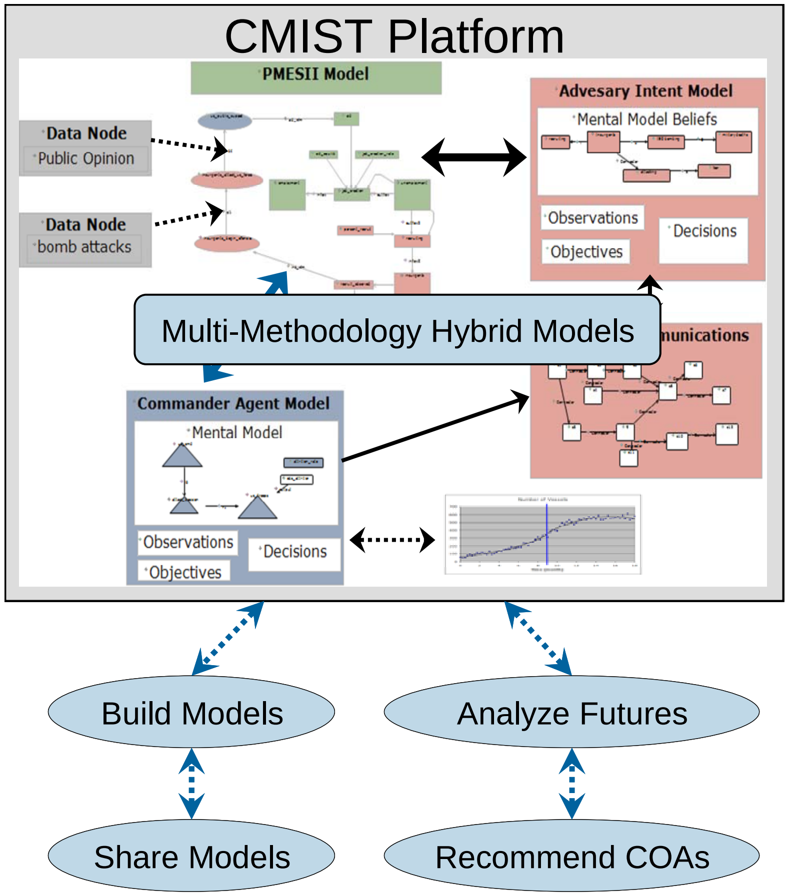

Figure 1. CMIST's hybrid modeling environment was extended to support advanced intent modeling via embedded simulation and Bayesian Knowledge Bases for inference of goals and beliefs.

2. SYSTEM ARCHITECTURE

CMIST is an Integrated Development Environment (IDE) for model authoring, integration, validation, and visualization. It is built on the versatile Eclipse framework, a widely used open source Java IDE. CMIST utilizes recent advances in model-driven architecture (MDA) to automate the process for integrating new modeling paradigms. The MDA process in turn is based on proven software engineering standards for model representation and exchange, such as Unified Modeling Language (UML) and eXtensible Markup Language (XML).

CMIST is designed with a two-stage architecture for distinct categories of users (see Figure 2):

1. A Developer's IDE for simulation and software developers to rapidly incorporate new simulation methodologies and tools, making them available for use in the Commander's IDE. The Developer's IDE provides the shared representation and common repository for model description and methodology characterization, such as timing, and method of computation. Eclipse's Graphical Modeling Framework (GMF) is used to design a representation for the graphical modeling primitives of a new methodology and automatically generate the Java source code to implement the data classes, controller, and editor palette for the methodology. After code generation, one need only implement a *mediator* to execute a specified increment of simulation time in a native simulation tool for the chosen methodology and to translate from that tool's native data model to CMIST's shared representation.

2. A Commander's IDE for commander and supporting staff to author models, integrate hybrid models via transform links, execute models over a specified time horizon, and debug the integrated models via intuitive visualization displays, including time series charts and map-based overlays.

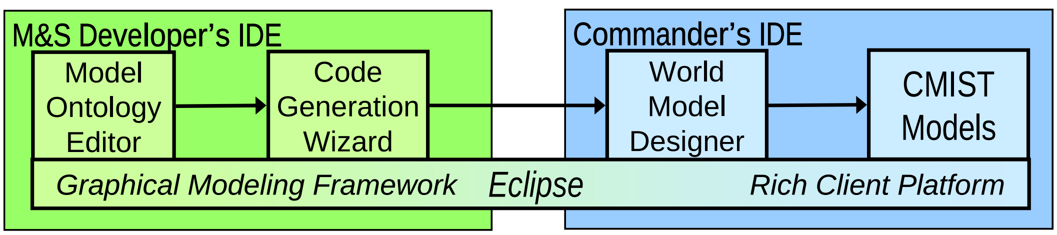

Figure 2. CMIST's two-tier architecture with IDEs for Modelers and for Developers

CMIST includes an extensible library for creating reusable data transforms that enable exchange of data between different modeling families. CMIST also provides multiple interaction patterns to synchronize multiple native simulations with disparate modeling paradigms, such as discrete time/event simulation, continuous-time simulation, and iterative solution methods such as Monte Carlo sampling.

Prior to running a model, CMIST uses the transform links to partition the overall model into fragments that each rely only on a particular methodology, as shown in Figure 3. It then compiles each fragment into a corresponding *native model* in an underlying third-party tool responsible for execution of that methodology. Finally, the CMIST simulation engine executes the model according to the chosen interaction pattern, for example, stepping time forward in each native model and then flowing outputs across transforms to the inputs of the next fragment.

The initial CMIST release successfully integrated three modeling methodologies and native tools, as shown in Figure 2:

- Probabilistic cause-effect modeling enabled by dynamic Bayesian network algorithms (DBN) [10] from AFRL's Operational Assessment Tool (OAT),
- System dynamics (SD) modeling based on [11] and [12], enabled by U.C. Berkeley's open source Ptolemy II framework, and
- Agent-based modeling enabled by Telecom Italia Lab's open Java Agent Development (JADE) framework.[1]

[1] http://jade.tilab.com/

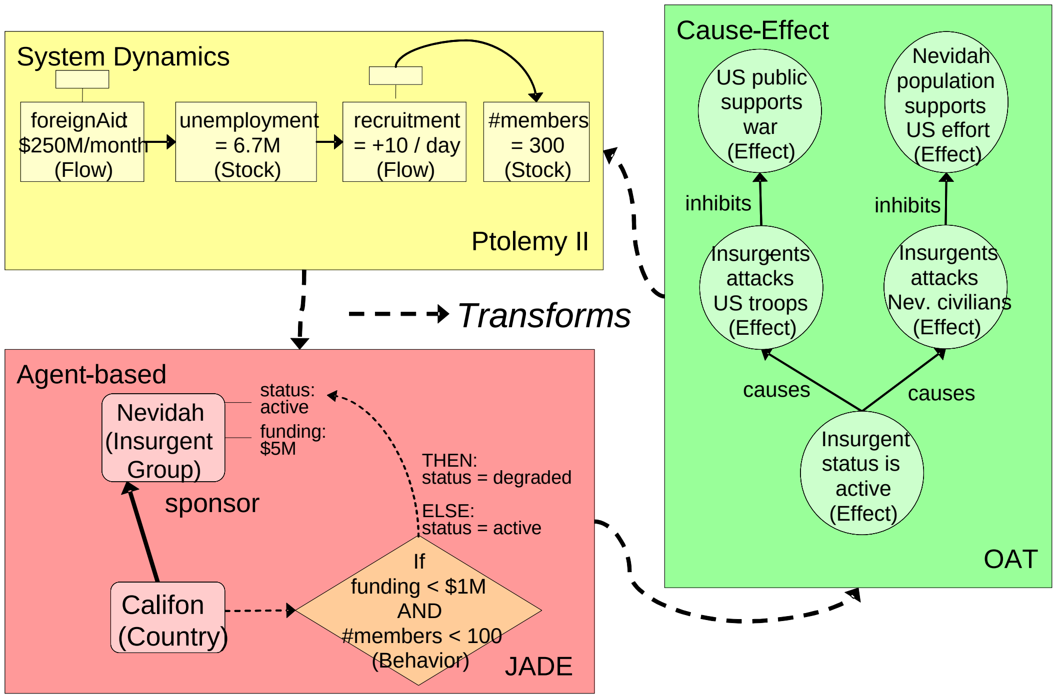

Figure 3. CMIST automatically compiles, executes, and integrates model fragments for multiple methodologies, including system dynamics, cause-effect, and agent-based modeling.

3. REACTIVE INTENT MODELING

This section describes a baseline Insurgent Growth model that is used as the backdrop for experimenting with reactive vs. proactive agent decision-making. The model is based on a larger-scale Military-Social-Information model investigating the effects of a non-kinetic PSYOPS campaign on major combat operations against a fictional adversary, Califon, which attempts to invade a neutral neighbor, Nevidah, in order to gain control of valuable mineral fields near the border between the two countries. For more details on this combined PSYOPS-combat model see [5].

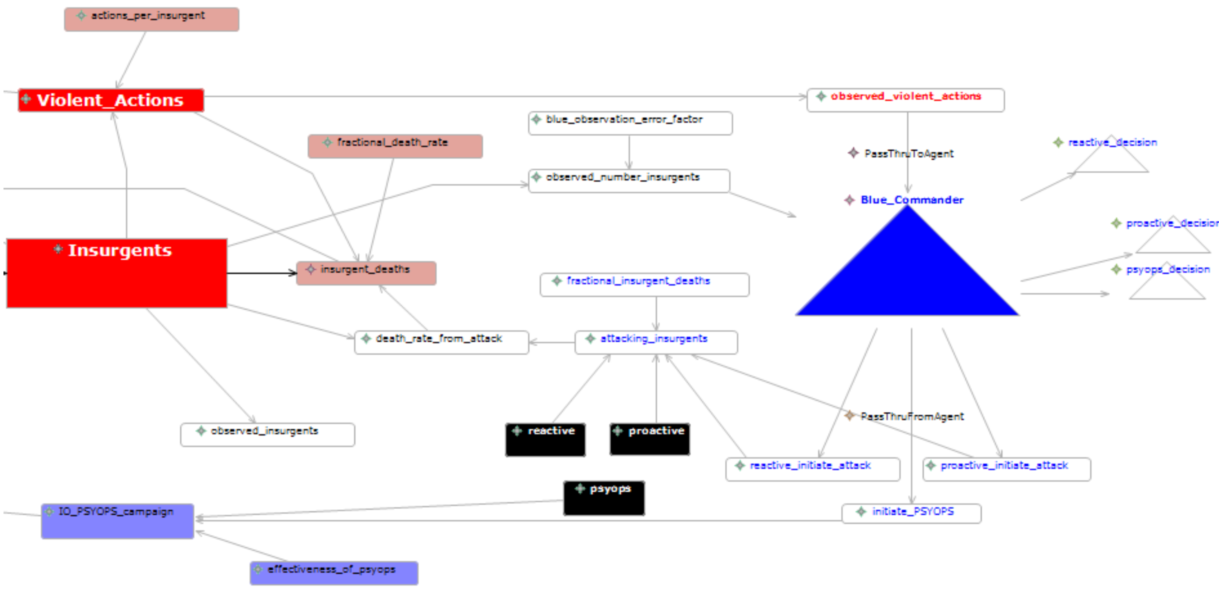

Figure 4. CMIST model fragment with switches for reactive vs. proactive counter-insurgency attack strategies.

The major dynamic of interest here is growth of Insurgents due to a positive feedback loop (not shown in Figure 4) that exploits fear among the general population to convert additional insurgents. This results in an S-shaped growth pattern (Figure 5, left). The Blue commander can observe the amount of violence and can observe an estimated quantity of insurgents. Based on this data, the Blue commander can reactively order an offensive attack to kill insurgents, or can proactively (preemptively) order the attack, and/or can mount a PSYOPS campaign to impact insurgent growth. Three switches in the model (shown in black) provide rapid access to these options: Reactive, Proactive, and Psyops (not discussed here but see [5]). When the Reactive switch is set to 1 and the other two are set to 0, the Blue_Commander agent bypasses its embedded model and utilizes built-in logic to initiate a counter-attack when observed_violent_actions exceeds a threshold of 50 incidents per day. The results are shown in Figure 5 (right). While the counter-attack eventually causes the rate of insurgent growth and violence to decrease, the peak violence of 75 is significantly higher than the desired threshold. Thus, by reacting only after the threshold was exceeded, the commander agent actually failed to adequately contain the level of violence. It is clear that a more proactive strategy is required, which we will demonstrate in the next section using embedded simulation.

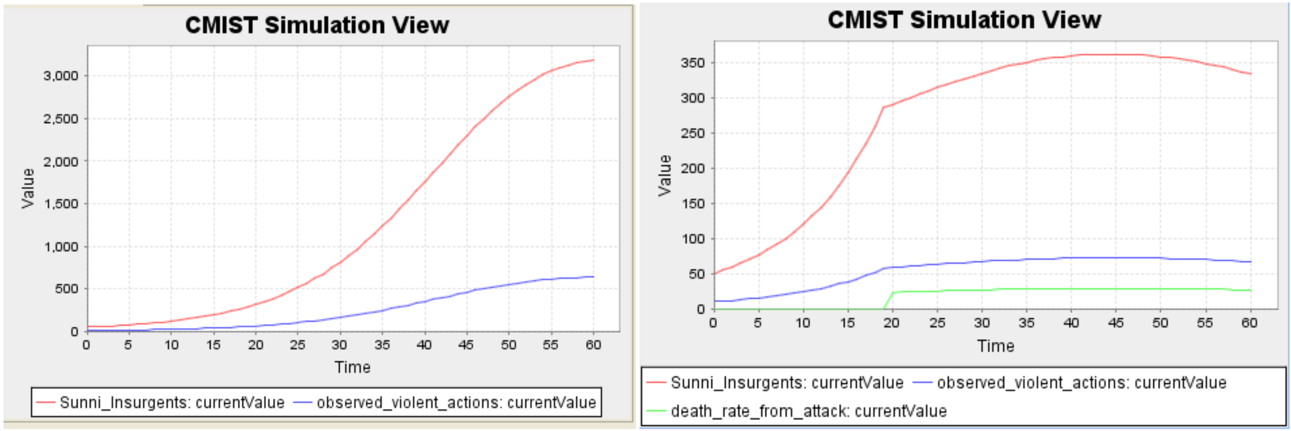

Figure 5. Insurgent growth and violence with no Blue intervention (left) vs. reactive Blue counter-attack (right).

4. EMBEDDED INTENT MODELING

4.1 Embedded Simulation Overview

In Section 0, we modeled commanders as reactive decision-makers. Here, we model commanders as proactive decision-makers. In order to be proactive, the agent must have some beliefs or projections of the *future* state of the world, rather than simply observing the *current* state of the world. To achieve this, we extended the JADE agent decision-making implementation to include a reference to a separate CMIST model. This model is embedded in the agent and is its representation of the world, a mental model of the "real" world. The *main model* refers to the primary model in CMIST that represents the real world situation (a portion is shown in Figure 4 above). This is the normal simulation model that would be built by an analyst to answer particular questions and forecast outcomes. The e*mbedded model* refers to a model that is built separately for use by an agent in the main model. Multiple agents may have their own model, or may share a model.

The simulation engine recursively executes embedded models. For each time-step in the main model, the embedded model *simulates for a user-defined number of time steps*. The simulation engine saves all output data for each individual embedded model run.

4.2 Proactive Intent Modeling

The proactive simulation allows the commander to forecast a single possible future based upon an embedded model and observations from the environment (Figure 6). The commander uses the observations to set initial parameter values in the embedded model, simulate the model for a fixed number of time steps, and retrieve the result. Here, the number of time steps is set to 20, allowing the commander to forecast 20 days into the future.

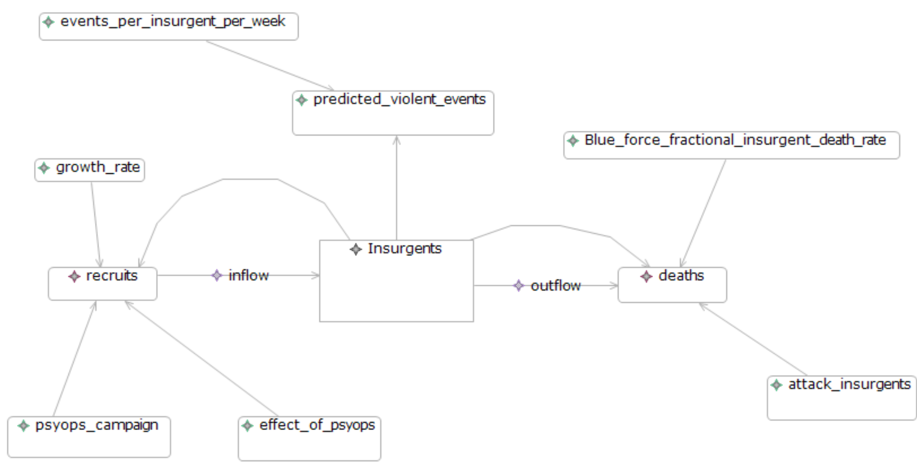

Figure 6. Commander's embedded mental model of the dynamics of insurgent growth

The core feature of the embedded model is a positive feedback loop in insurgent recruiting. When the blue **psyops_campaign** and **attack_insurgent** variables are false, the number of insurgents grows exponentially at a rate specified by the variable **growth_rate.** When **psyops_campaign** is true, the recruiting rate lessens by the scale factor **effect_of_psyops.** When **attack_insurgents** is true, the negative feedback from the SD flow node **deaths** counteracts the insurgent growth.

The commander agent's forecast from this embedded model is compared to a desired threshold value for amount of violence (a property set within the agent). When the *forecasted* violence exceeds the threshold, the commander orders a mission to attack the insurgents.

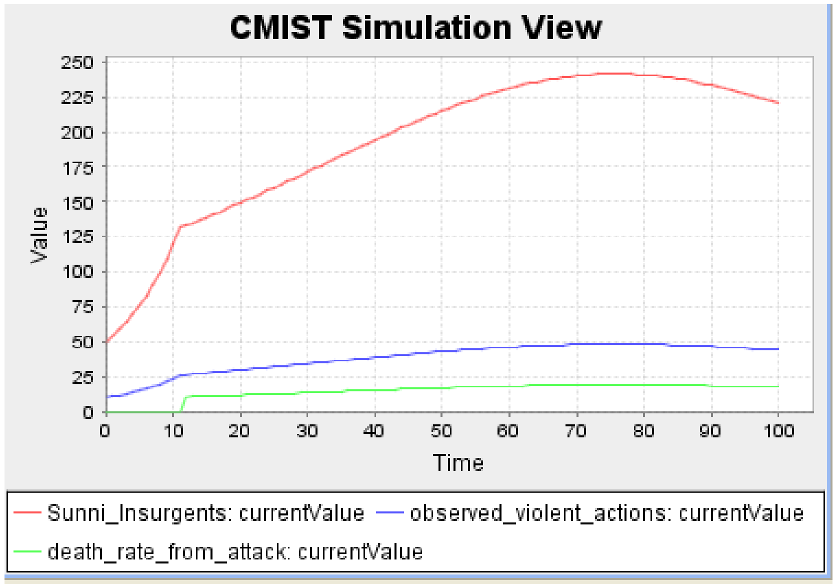

Figure 7. Proactive offensive to attack insurgents

Figure 7 shows the main model results when the proactive switch is on and the other switches are off. Note how the attack starts much earlier (Day 12 as opposed to Day 20) and leads to significantly fewer insurgents (240 instead of 360) and less violence at their peak than in the reactive scenario. Note also that the death rate of insurgents from the blue attack is smaller, because the attack commenced when there were fewer insurgents and our main model applies a fractional death rate multiplier to the number of insurgents. This proves to be a much more effective strategy than the reactive approach, since the Blue commander actually achieved its goal of keeping violence rate under the desired threshold of 50 incidents per day.

5. ADVERSARY INTENT MODELING WITH BAYESIAN KNOWLEDGE BASES

While our new embedded simulation capability allows an agent to more proactively match intended actions to intended outcomes, it still fails to adequately capture the *beliefs and goals* that motivate the agent's intent. Furthermore, CMIST lacked a mechanism for reasoning over observed evidence to infer which goals are currently being pursued. To address this gap, we enlisted the help of Dr. Eugene Santos, Jr. and his team at Dartmouth College, to integrate a new modeling family, Bayesian Knowledge Bases (BKBs). We chose BKBs because it preserves both a strong semantics and an intuitive organization of knowledge, manages uncertainty between events [6], can be constructed easily based on incomplete information [7] and is computational efficient due to its compact structure [13]. This extension on intent modeling also facilitates our modeling of commanders' intent including their behavior and beliefs. More detail about adversarial intent inferencing and dynamic wargaming applications can be found in [8] and [9].

5.1 BKB Overview

Like CMIST's existing DBN modeling family, which is used for forward inference in cause-effect models, BKBs are abstractions over normal Bayesian Networks (BNs) [14], requiring only one probability estimate per node. But unlike DBNs, BKBs retain the expressiveness of BNs in that each random variable may have more than just true or false states. BKBs feature two types of nodes. Instantiation nodes or "I-nodes," depicted as text-boxes in Figure 8 (b), express whether the node's random variable is in a particular state. An I-node is uniquely defined by one state of a random variable. Thus there can be many different I-nodes that represent various states of a single random variable. This is in contrast to a Bayesian Network where each node represents a full random variable along with all of its states. Instead of causal links, BKBs feature support nodes, or "S-nodes," depicted as black dots that encode the uncertainty of the causal relationships between I-nodes. The input to an S-node can be zero or more I-nodes, the output is exactly one I-node, and a weight is given to each S-node. The weight of an S-node is the conditional probability of its head I-node given that its tail I-nodes are known to be true. This sparse "if-then" structure provides the knowledge engineering advantages of forward-chaining expert systems, including soundness of inference in the absence of complete knowledge. BKBs can also capture cyclical knowledge structures not allowed in BNs or DBNs. Finally, BKBs are well-suited to direct closed form inference rather than sampling. BAE Systems and Dartmouth have also shown that random-variable level acyclic BKBs can also be mapped to a full BN representation [ref], allowing for fast inference in third-party BN tools such as Norsys's Netica.®

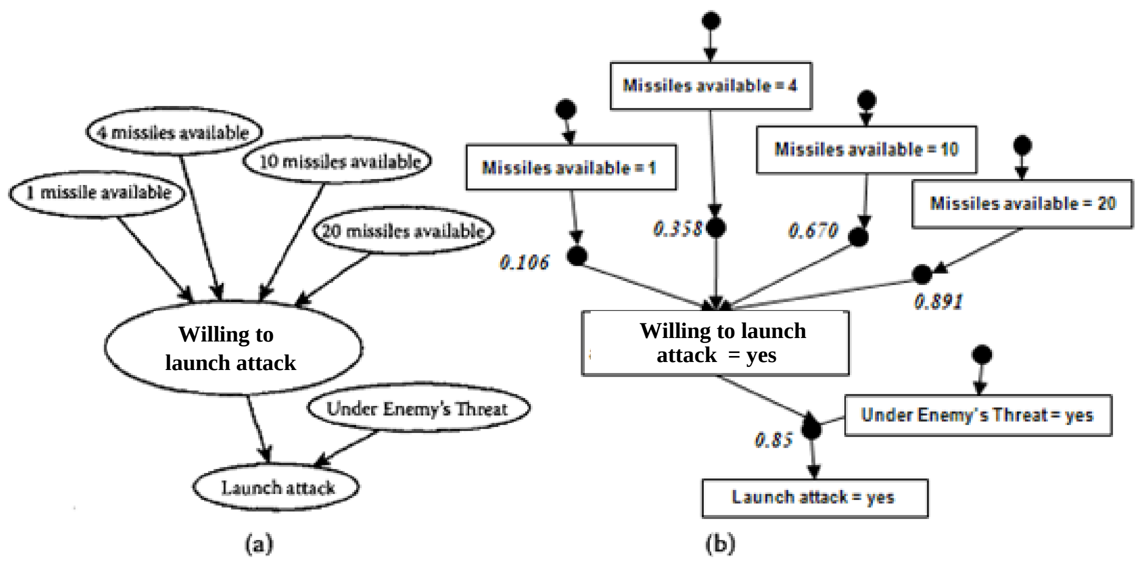

Figure 8. A probabilistic model using a BN or DBN (a) versus a BKB (b) (Source: [6]).

The main function of BKBs is to do probabilistic inferencing so as to find the marginal probability of each random variable and answer questions regarding the most probable state of the world. The modeled world can be updated based on evidence which is observed about states of the random variables. In other words, evidence sets the states of random variables that are known to be true (to a value of 1.0), and the inferred state of the world is the most likely world that supports the evidence. The function of probabilistic inferencing includes prediction and explanation. Prediction is achieved by extending the current evidence forward to the unknown state of the world, while explanation is to determine the cause of current state by extending the known evidence backward to hypotheses.

5.2 BKB Intent Model

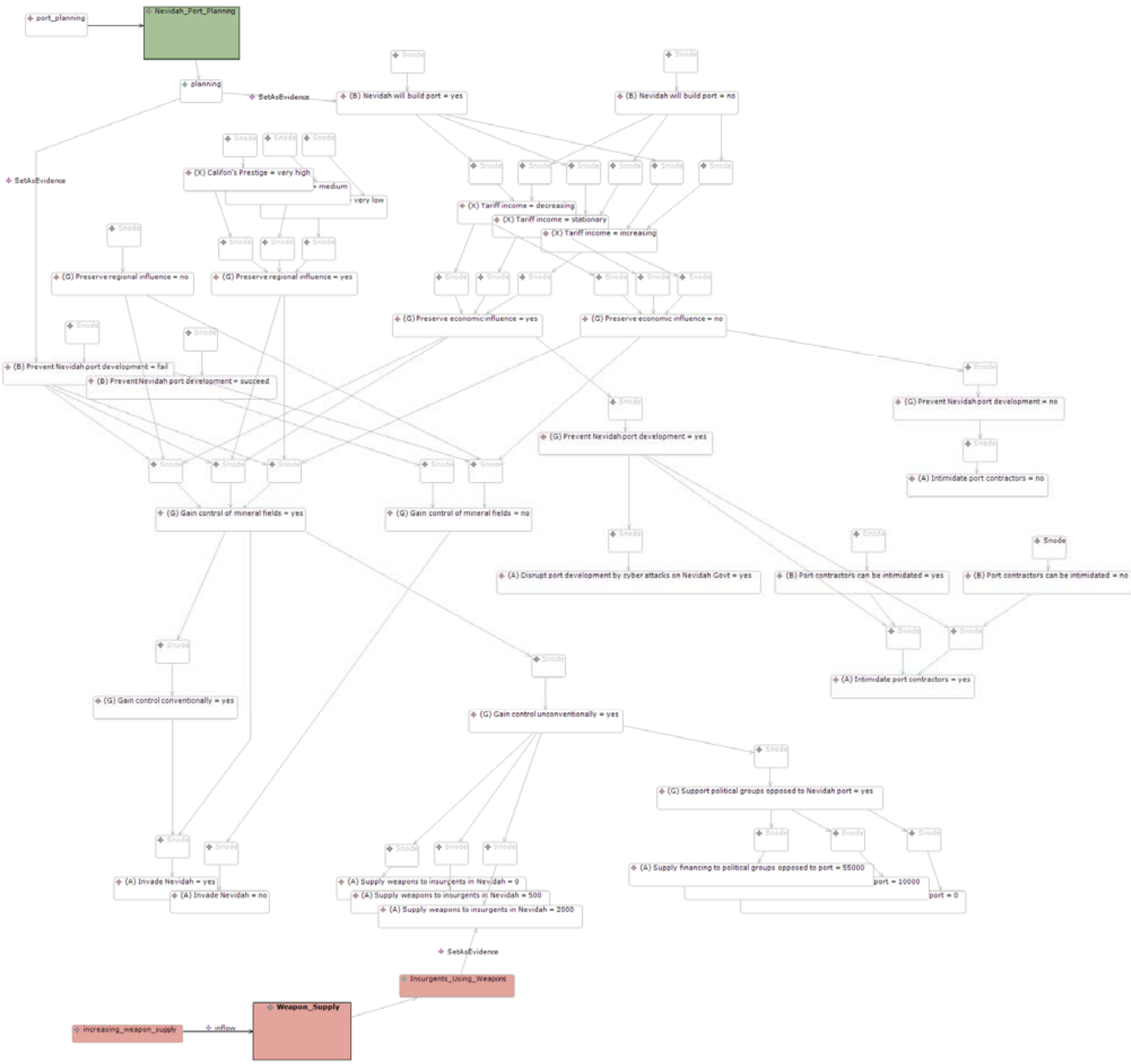

Figure 9. BKB intent model for Califon.

This section covers a short description of a BKB fragment developed by BAE Systems and Dartmouth to model Califon's intent (shown in Figure 9). The model includes four main types of I-Nodes, in keeping with an intent modeling convention developed by Dartmouth:

- (B) Belief – What the entity believes about others
- (X) Axiom – What the entity believes about itself
- (G) Goal – What results the entity wants to achieve
- (A) Action – How the entity will carry out its tasks

The main intent drivers in this model come from the belief "(B) Nevidah will build a port." Califon's axiom "(X) Tariff income" is that the Nevidah port will either decrease income (most likely), maintain the same income, or increase income (least likely). The amount of Califon tariff income will influence their goal "(G) Preserve economic influence." This goal then drives at least two different goal-action structures: 1) to gain control of the mineral fields, and 2) to prevent Nevidah's port development.

Califon actions occur near the bottom of the diagram. When values change in the CMIST model to reflect changes in the real world, they serve as evidence that can be used to influence the known state of BKB nodes (usually actions, but here we also set evidence on a Califon belief). Since CMIST runs the BKB inference engine for each time-step in the main simulation, when new evidence is received, the probabilities in the BKB model are adjusted to reflect the most-likely explanation, setting the most probable nodes to true (1.0). Collectively, the nodes inferred to be true represent one or more causal paths from upstream I-Nodes to the observed I-Nodes.

We will demonstrate this inference capability by setting evidence on different combinations of I-Nodes in the model. In addition to setting evidence directly on an I-Node, our model can transmit evidence from a System Dynamics variable using a *transform* link. When the SD stock Nevidah_Port_Planning's value (the shaded box shown at the top of the model) exceeds 100, the transform labeled "Set as Evidence" activates and sets the I-Node "(B) Nevidah will build port = yes" to true. In addition, we have also directly set evidence on the I-Node "(A) Intimidate Port Contractors = yes" to true, since this represents a low-risk tactic used by Califon. All other I-Nodes have no evidence set, which is the default. Upon running the model for eight time-steps, we observe in Figure 10 (left) that the resulting value of this I-Node is indeed true (1.0) from Day 1 forward (values at time-step 0 are disregarded because it is an initialization step). Furthermore, CMIST detects that the stock Nevidah_Port_Planning exceeds 100 on Day 3, causing the I-Node "(B) Nevidah will build port = yes" to be true from Day 4 forward (information takes one time-step to flow across transform links).

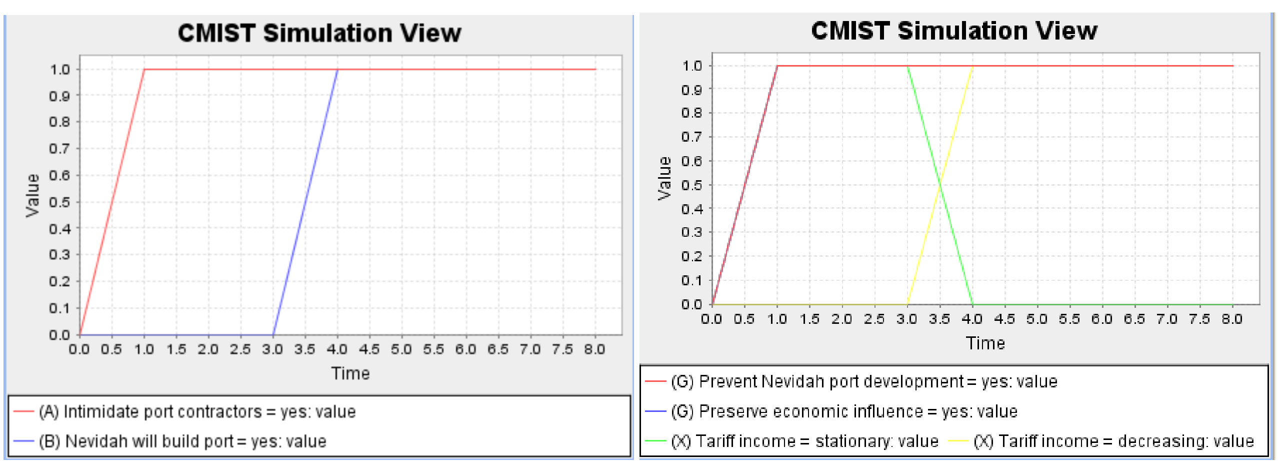

Figure 10 (left) shows the timing of evidence in the model of Califon's intimidation of contractors and Nevidah's port construction, and (right) shows the BKB's inference about Califon's intermediate goals and axions.

Figure 10 (right) shows additional goals and axioms inferred by the BKB as the most probable explanation for the evidence. Even on Day 1, before Nevidah as begun building the port, Califon's intimidation of port contractors is evidence of Califon's goals "Prevent Nevidah port development = yes" and "Preserve economic influence = yes." However, initially Califon's axiom "Tariff income = stationary" is true because it has not seen evidence that Nevidah is building the port. This changes on Day 3, when its value changes to false and the value of axiom "Tariff income = decreasing" changes to true.

Now, we examine another aspect of the intent model describing Califon's goal to gain control of the disputed mineral fields across the Nevidah border. The SD stock **Weapon_Supply** at the bottom of the model increases by 500 each time-step. The evidence transform link from the SD variable **Insurgents_Using_Weapons** to the BKB action "(A) Supply weapons to insurgents in Nevidah = 2000" will activate when the stock's value exceeds 2000. As shown in Figure 11 (left), this action node becomes true from Day 5 forward. The BKB then computes the most probable causal explanation, working from low-level to high-level goals. The model infers that the following goals are true, as shown in Figure 11 (right):

- (G) Gain control unconventionally = yes
- (G) Gain control of mineral fields = yes
- (G) Preserve regional influence = yes

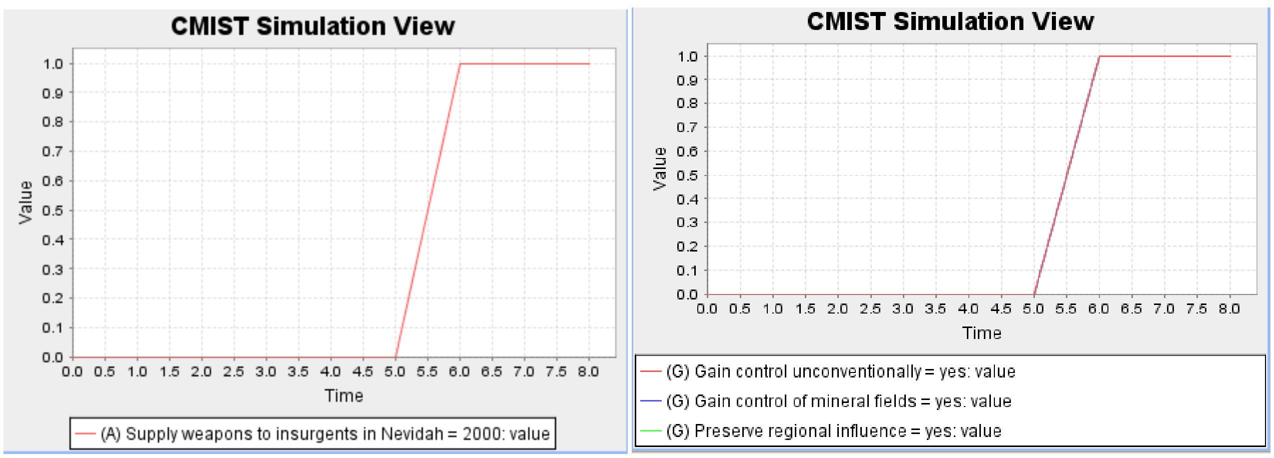

Figure 11. When Califon's weapons supply to insurgents exceeds 2000, the BKB infers Califon's intention to gain control of mineral fields using unconventional means.

6. TEMPORAL BAYESIAN KNOWLEDGE BASES

In real-time complex systems, the interactions of processes change with time. Especially in the battlespace, the change of strategies in response to the passage of time is as important as other stimuli. Therefore, our system needs a unified model to capture both the semantics for uncertainty and the semantics for time. Our initial integration of BKB within CMIST's simulation engine described above allowed the BKB to react to changes in evidence from other modeling families at different times, but the BKB itself did not contain a representation of time. Under this effort, Dartmouth has developed an extension of BKB that incorporates temporal constraints following the design of a temporal network in [16]. This was done without sacrificing BKB's expressiveness (encoding of multi-valued states) or their efficiency (use of closed form inference rather than sampling). This new Temporal BKB (TBKB) extension allows users to specify a time interval (a, b) on each I-node, where "a" denotes the start time, and "b" denotes the duration, and temporal dependencies between I-nodes on S-nodes. The time dependency is represented by one of the three time relations: equals (=), precedes (<) and follows (>), and by the latency between the start times of two I-nodes. During the inferencing of TBKB, the observed time interval together with the state can be submitted as evidence. However, it is sometimes unattainable to obtain the complete temporal information of an event. Therefore, we also allow incomplete information in the evidence. Figure 12 shows a TBKB fragment depicting timing of goals and beliefs and delayed secondary goals and actions. The start and end times on I-Nodes represent timing of causes. The I-Node is assumed to have the specified probability only during this interval; at all other times its probability is unknown by default. The S-Node delays allow specification of timing of both *disjunctive* (OR) and *conjunctive* (AND) effects. The pair of I-Nodes G1=Yes and B2=No, and the pair of B1=Yes and B2=Yes each have a separate child S-Node with its own conditional probability and time delay, an example of disjunctive temporal causes. During the delay period the model behaves as if there is no S-Node connection. After the delay, the usual probability semantics for S-Nodes and I-Nodes hold. The I-Nodes G2=Yes and G3=Yes share a single child S-Node, indicating a conjunctive causal influence; the child I-Node (A1) is not affected until at least 10 days after both G2 and G3 become true. This ability to model temporal aspects of an agent's intent provides a more expressive representation for enemy courses of action, including sequencing of tasks and delayed, cascading effects.

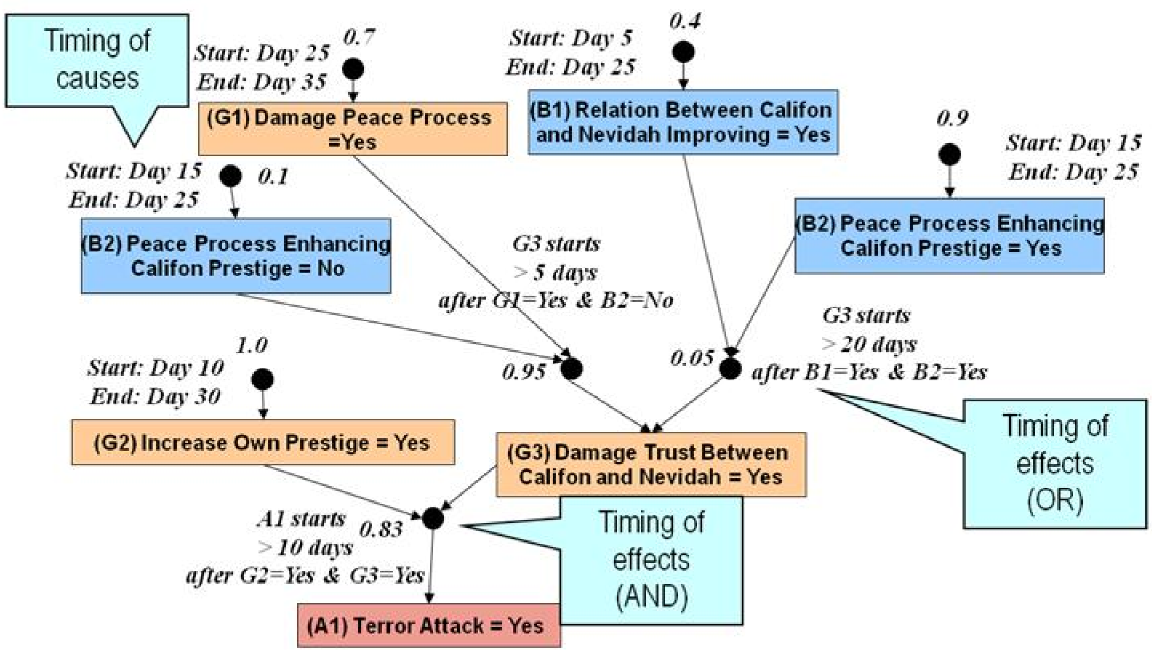

Figure 12. Example TBKB model fragment depicting timed I-Nodes and delays on S-Nodes.

7. FUTURE WORK

Our current research plans for CMIST include several new capabilities. BAE Systems and Dartmouth are currently collaborating on a new Colored Petri Net (CPN) modeling family [13], based on an open-source Petri net modeling tool called Platform-Independent Petri Net Environment (PIPE).[2] We have extended PIPE to support colored tokens with deterministic time delays and stochastic prioritization of transitions. CPNs are particularly well suited to modeling information flow within dynamic processes. We have begun CPN modeling of several relevant Air Force operational processes, including a Time Critical Targeting cell, which will be used in our Califon scenario to strike against enemy Surface-to-Air Missile threats. In addition to CPNs, we plan to extend CMIST with new nodes and attributes for capturing source metadata used to inform model structure and initialization. Building such documentation directly into models will facilitate validation of models and help explain the rationale behind a model's design to decision-makers.

[2] http://pipe2.sourceforge.net/

8. CONCLUSIONS

We have integrated into CMIST two new techniques for adversary intent modeling that are significantly more sophisticated than the methods available in previous agent-based and DBN models. First, embedded simulation provided a means of encoding a *dynamic* internal model within an agent. Using a notional counter-insurgency model, we showed how this can be used to trigger proactive decision-making based on the agent's forecast of future state from current trends. Second, Bayesian Knowledge Bases provided a richer means of modeling an agent's intent in terms of its axioms and beliefs about itself and others, its primary and secondary goals, its planned actions, and the causal relationships among these concepts. Unlike the existing DBN modeling family, BKBs provided a means of inferring an agents goals and beliefs from observed evidence, by computing the most probable explanation. Furthermore, transform links enable other modeling families such as System Dynamics to interact with BKB fragments by setting evidence at certain times during execution, causing the BKB to update its most probable explanation dynamically. Temporal BKBs, currently in progress, will provide an even richer dynamic intent representation, modeling scheduled actions and delays between causes and effects. Collectively, these new capabilities provide a powerful intent modeling capability for warfighters and analysts to better understand their adversaries and forecast possible futures.

ACKNOWLEDGMENTS

This material is based upon work supported by the United States Air Force Research Laboratory under Contract No. FA8750-08-C-0101. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

©REFERENCES

[1] Biggie, J., "Operational Net Assessment," Briefing to Military Operations Research Society, Decision Aids meeting, USJFCOM-J9, Nov. 19, 2003.

[2] Bross, P. J., "Measuring the Will to Fight in Simulation," *73rd Military Operations Research Society Symposium*, West Point, NY, 2005.

[3] Pioch, N. J., Lofdahl, C., Sao Pedro, M., Krikeles, B., and Morley, L. "An Integrated Development Environment for PMESII Model Authoring, Integration, Validation, and Debugging," *SPIE Defense and Security – Modeling and Simulation for Military Operations*, Orlando, FL, April, 2007.

[4] Pioch, N. J., "Commander's Model Integration and Simulation Toolkit," BAE Systems Advanced Information Technologies, Final Technical Report TR-2067, Oct. 2007.

[5] Pioch, N. J., Melhuish, J., and Cook, M., "A Case Study in Integrated PMESII Modeling and Simulation," *76th Military Operations Research Society Symposium*, Briefing for Working Group 1 (Strategic Ops) & 8 (Information Ops), New London, CT, 2008.

[6] Santos, E., Jr, Santos, E. S., "A framework for building knowledge-bases under uncertainty", in Journal of Experimental and Theoretical Artificial Intelligence, Vol: 11, pp. 265-286, 1999.

[7] Santos, E., Jr. and Zhao, Q., "Adversarial Models for Opponent Intent Inferencing," in Adversarial Reasoning: Computational Approaches to Reading the Opponents Mind (Eds. A. Kott and W. McEneaney), pp 1-22, Chapman & Hall/CRC Computer and Information Science Series, Chapman & Hall/CRC:Boca Raton, 2006.

[8] Santos, E., Jr., Zhao, Q., Pratto, F., Pearson, A. R., McQueary, B., Breeden, A., and Krause, L., "Modeling Multiple Communities of Interest for Interactive Simulation and Gaming: The Dynamic Adversarial Gaming Algorithm Project," *Proceedings of the SPIE: Defense & Security Symposium*, Vol: 6564, Orlando, FL, 2007.

[9] Lehman, L. A., Krause, L. S., Gilmour, D. A., Santos, E., Jr., and Z., Qunhua "Intent Driven Adversarial Modeling," *Proceedings of the Tenth International Command and Control Research and Technology Symposium: The Future of C2*, McLean, VA, 2005.

[10] Gossink, D., Lemmer, J. "Recursive Noisy OR – A Rule for Estimating Complex Probabilistic Actions," IEEE Transactions on Systems, Man, and Cybernetics, Part B, December 2004.

[11] Forrester, J. "System Dynamics and the Lessons of 35 Years," MIT Sloan School of Management, Technical Report D-4224-4, 1991.

[12] Sterman, J. *Business Dynamics: Systems thinking and modeling for a complex world*. Boston: McGraw-Hill, 2000.

[13] Rosen, T., Shimony, S. E., and Santos, E., Jr., "Reasoning with BKBs – Algorithms and Complexity," in Annals of Mathematics and Artificial Intelligence, Vol: 40(3-4), pp. 403-425, 2004.

[14] Pearl, J. *Probabilistic Reasoning in Intelligent Systems*. San Francisco, CA: Morgan Kaufmann Publishers, 1988.

[15] Hunter, D., Krause, L., and Santos, E., Jr. "Maximum Entropy Inference for Bayesian Knowledge-bases," Unpublished Technical Report, 2008.

[16] Santos, E., Jr. and Young, J. D., "Probabilistic Temporal Networks: A unified framework for reasoning with Time and Uncertainty," in International Journal of Approximate Reasoning, Vol: 20, pp. 263-291, 1999.

[17] Jensen, K. (1992). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. EATCS Monographs on Theoretical Computer Science. Vol. 1. Springer-Verlag, New York, NY.